AMENDMENTS TO THE CLAIMS:

Kindly amend claims 1, 3, 7, 8,13-15, 21, 24-27, 33 and 34, as shown below.

This listing of claims will replace all prior versions and listings of claims in the Application:

Claim 1 (currently amended): A method for recovering an application from a runtime fault, the method comprising steps of:

receiving an exception caused due to a runtime fault in a thread executing [[an]] a C++ based application;

dispatching the exception to an exception handler;

trapping the exception before the exception reaches the exception handler when the exception handler is a top level exception handler which terminates the application;

translating the trapped exception into [[an]] a C++ exception that the C++ based application is capable of handling; and

continuing execution of the application.

Claim 2 (original): The method recited in claim 1 further comprising a step of terminating the thread that caused the exception.

Claim 3 (currently amended): The method recited in claim 1, wherein the dispatching step comprises steps of:

determining a corresponding exception handler to which the exception is to be dispatched;

dispatching the exception to the corresponding exception handler when the corresponding exception handler exists; and

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

dispatching the exception to a top level dispatcher ~~is the corresponding exception handler~~ when no corresponding exception handler exists.

Claim 4 (previously presented): The method recited in claim 1 further comprising a step of dispatching the trapped exception to a trapped exception handler.

Claim 5 (original): The method recited in claim 4 further comprising a step of terminating the thread when the trapped exception handler is not capable of resolving the trapped exception.

Claim 6 (original): The method recited in claim 5, wherein the continuing step allows continuing execution of the application after the thread is terminated.

Claim 7 (currently amended): The method recited in claim 1 wherein:

the translating step translates the trapped exception into [[an]] the C++ exception which is able to be resolved by ~~a lower level~~ an application defined C++ exception handler, and

the method further comprises the step of determining if there is ~~a lower level~~ an application based C++ exception handler which is capable of resolving the translated exception.

Claim 8 (currently amended): The method recited in claim 7 further comprising a step of terminating the thread that caused the exception when there is no ~~lower level~~ C++ based exception handler which is capable of resolving the translated exception.

Claim 9 (previously presented): The method recited in claim 2 further comprising a step of logging state information representing the state that the application was in before occurrence of the exception caused the termination of the thread.

Claim 10 (original): The method recited in claim 9 further comprising a step of forwarding the logged information to a remote database over a computer network.

Claim 11 (original): The method recited in claim 10 further comprising steps of:

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

receiving a recommendation from the remote database; and

informing the recommendation to the user.

Claim 12 (original): The method recited in claim 9 further comprising a step of forwarding a bug report to a bug report centre over a computer network.

Claim 13 (currently amended): A method for recovering an application from a runtime fault in a thread, the application being executed under an operating system having one or more low level exception handlers and a top level exception handler, the method comprising steps of:

trapping an exception which is dispatched to the top level exception handler before the exception reaches the top level exception handler, the exception being caused due to a runtime fault in a thread executing [[an]] a C++ based application, a default action of the top level exception handler being to terminate the application upon receipt of exceptions;

translating the trapped exception into [[an]] a C++ exception that the C++ based application is capable of handling; and

continuing execution of the application.

Claim 14 (currently amended): The method recited in claim 13 wherein:

the translating step translates the trapped exception into [[an]] the C++ exception which is able to be resolved by ~~a lower level~~ an application based C++ exception handler, and

the method further ~~comprising~~ comprises a step of determining if there is ~~a lower level~~ an application based C++ exception handler which is capable of resolving the translated exception.

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

Claim 15 (currently amended): The method recited in claim 14 further comprising a step of terminating the thread that caused the exception when there is no ~~lower-level~~ C++ based exception handler which is capable of resolving the translated exception.

Claim 16 (original): The method recited in claim 13 further comprising a step of terminating the thread that caused the exception.

Claim 17 (previously presented): The method recited in claim 16 further comprising a step of logging state information representing the state that the application was in before occurrence of the exception caused the termination of the thread.

Claim 18 (original): The method recited in claim 17 further comprising a step of forwarding the logged information to a remote database over a computer network.

Claim 19 (original): The method recited in claim 18 further comprising steps of:

receiving a recommendation from the remote database; and

informing the recommendation to the user.

Claim 20 (original): The method recited in claim 17 further comprising a step of forwarding a bug report to a bug report centre over a computer network.

Claim 21 (currently amended): An application recovery system for recovering an application from a runtime fault, the application recovery system comprising:

an exception dispatcher for receiving an exception caused due to a runtime fault in a thread executing [[an]] a C++ based application, and dispatching the exception to an exception handler;

an exception trapper for trapping the exception before the exception causes termination of the application;

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

an exception translator for translating the exception trapped by the exception trapper into [[an]] a C++ exception that the C++ based application is capable of handling; and

an executor for continuing execution of the application.

Claim 22 (original): The application recovery system as claimed in claim 21, wherein the exception trapper has a thread terminator for terminating the thread that caused the exception.

Claim 23 (previously presented): The application recovery system as claimed in claim 21, wherein the exception trapper is provided in place of a top level exception handler which terminates the application.

Claim 24 (currently amended): An application recovery system for recovering an application from a runtime fault caused in a thread, the application running under an operating system having an exception dispatcher, one or more low level exception handlers and a top level exception handler which terminates the application, the application recovery system comprising:

an exception trapper placed between the exception dispatcher and the top level exception handler for trapping an exception before the exception reaches the top level exception handler, the exception being caused due to a runtime fault in a thread executing [[an]] a C++ based application;

an exception translator for translating the trapped exception into [[an]] a C++ exception that the C++ based application is capable of handling; and

a trapped exception handler for handling the trapped exception.

Claim 25 (currently amended): The application recovery system recited in claim 24, wherein the trapped exception handler comprises a thread terminator for terminating the thread

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

when there is no ~~lower level~~ C++ based exception handler that is capable of handling the translated exception.

Claim 26 (currently amended): The application recovery system recited in claim 24, wherein the trapped exception handler comprises:

an exception handler selector for determining if ~~a lower level~~ an application based C++ exception handler is capable of resolving the exception translated by the exception translator.

Claim 27 (currently amended): The application recovery system recited in claim 26, wherein the trapped exception handler further comprises a thread terminator for terminating the thread when there is no ~~lower level~~ C++ based exception handler that is capable of handling the translated exception.

Claim 28 (original): The application recovery system recited in claim 27, wherein the trapped exception handler further comprises a state restorer for restoring the state that the application was in before the fault occurred to continue the execution of the application.

Claim 29 (original): The application recovery system recited in claim 24 further comprising a state information logger for logging information of the state that the application was in before the fault occurred.

Claim 30 (original): The application recovery system recited in claim 29 further comprising a query generator for generating a query including the state information to query a recommendation from a remote database over a computer network.

Claim 31 (original): The application recovery system as claimed in claim 30 further comprising a user advisor for receiving a recommendation from the remote database and informing the user of the recommendation.

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

Claim 32 (original): The application recovery system as claimed in claim 30 wherein the query generator has a bug report generator for forwarding a bug report with the state information to a bug report centre.

Claim 33 (currently amended): A computer readable memory element storing the instructions or statements for use in the execution in a computer of a method for recovering an application from a runtime fault, the method comprising steps of:

receiving an exception caused due to a runtime fault in a thread executing [[an]] a C++ based application;

dispatching the exception to an exception handler;

trapping the exception before the exception reaches the exception handler when the exception handler is a top level exception handler which terminates the application;

translating the trapped exception into an exception that the application is capable of handling; and

continuing execution of the application.

Claim 34 (currently amended): Electronic signals for use in the execution in a computer of a method for recovering an application from a runtime fault, the method comprising steps of:

receiving an exception caused due to a runtime fault in a thread executing [[an]] a C++ based application;

dispatching the exception to an exception handler;

trapping the exception before the exception reaches the exception handler when the exception handler is a top level exception handler which terminates the application;

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

translating the trapped exception into [[an]] a C++ exception that the C++ based application is capable of handling; and

continuing execution of the application.

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567